CLAIMS

1. A method for computing ranks in a linked database, the method comprising:

obtaining a local rank vector associated with a selected subset of nodes in the linked database, wherein each component of the local rank vector represents a local rank of a node in the selected subset of nodes;

obtaining a block rank vector associated with the linked database, wherein each component of the block rank vector represents a block rank of a subset of nodes in the linked database, wherein the subset is one of a plurality of subsets of nodes defined by a partition of the nodes in the linked database;

selecting a component of the block rank vector corresponding to the selected subset of nodes;

selecting a component of the local rank vector corresponding to a selected node in the selected subset of nodes;

combining the selected component of the block rank vector and the selected component of the local rank vector to obtain a global rank for the selected node.

2. The method of claim 1 wherein obtaining the local rank vector comprises receiving the local rank vector from a computer that calculated the local rank vector.

3. The method of claim 1 wherein obtaining the local rank vector comprises selecting components of a preexisting global rank vector.

4. The method of claim 1 wherein obtaining the local rank vector comprises forming a local link matrix comprising link weights between nodes of the selected subset and computing the local rank vector from the local link matrix.

5. The method of claim 1 further comprising classifying the nodes of the linked database into subject classes and creating the partition of the nodes into the plurality of subsets in accordance with the subject classes.

6. The method of claim 1 further comprising obtaining a plurality of local rank vectors associated with the plurality of subsets.

7. The method of claim 1 wherein obtaining the block rank vector comprises forming a reduced link matrix for the linked database and computing the block rank vector from the reduced link matrix.

8. The method of claim 7 wherein forming the reduced link matrix comprises using a set of preference weights associated with the subsets to alter elements of the reduced link matrix so that the block rank vector is customized in accordance with the preference weights.

9. The method of claim 1 further comprising calculating a final rank from the global rank using an iterative link-based ranking technique.

10. The method of claim 1 further comprising using the global rank to determine an order of presentation of the selected node among other nodes.

11. A method for computing a rank value for a node in a linked database, the method comprising:

partitioning nodes of the linked database into K subsets according to a classification of the nodes;

computing K local rank vectors for the K subsets of the nodes;

computing a block rank vector from a KxK reduced link matrix;

computing a global rank vector from the local rank vector and the block rank vector;

and

selecting a component of the global rank vector corresponding to the node to obtain the rank value for the node.

12. The method of claim 11 wherein computing the K local rank vectors comprises arranging a link matrix for the linked database into a block-diagonal form corresponding to the partition of the nodes into subsets; forming K local link matrices from blocks of the link matrix, and computing the K local rank vectors from the K local link matrices.

13. The method of claim 11 wherein computing the K local rank vectors comprises executing a link-based ranking algorithm on a local link matrix.

14. The method of claim 13 wherein the link-based ranking algorithm comprises calculating a principal eigenvector of the local link matrix.

15. The method of claim 13 wherein the link-based ranking algorithm comprises performing a singular value decomposition of the local link matrix.

16. The method of claim 13 wherein the link-based ranking algorithm comprises forming a vector representing the row sums or column sums of the matrix.

17. The method of claim 11 wherein computing the K local rank vectors comprises dividing a preexisting global rank vector into K parts.

18. The method of claim 11 wherein computing the block rank vector comprises forming a reduced link matrix whose elements represent links between the subsets of the partition, and calculating the block rank vector from the reduced link matrix.

19. The method of claim 18 wherein forming the reduced link matrix comprises computing a block link weight between a first block and a second block by adding together weights of links from nodes in the first block to nodes in the second block.

20. The method of claim 18 further comprising customizing the reduced link matrix using a set of preference weights associated with the subsets.

21. The method of claim 11 wherein computing the global rank vector from the local rank vector and a block rank vector comprises:

computing an estimated global rank vector from the local rank vector and a block rank vector;

computing the global rank vector from the estimated global rank vector using an iterative link-based ranking technique.

22. The method of claim 11 wherein computing the K local rank vectors is performed at K distributed computers, and wherein computing the global rank vector is performed at a central computer.

23. The method of claim 11 wherein the linked database is a distributed collection of hypertext documents and the classification of the nodes is based on URL addresses of the nodes.

24. The method of claim 11 wherein the classification of the nodes is a predetermined subject classification of documents in the linked database.

25. The method of claim 11 wherein the classification of the nodes is computationally determined from a link structure of the linked database.

26. The method of claim 11 wherein the classification of the nodes is computationally determined from a similarity of content associated with nodes.

27. A method for computing a rank value for a block of nodes in a linked database, the method comprising:

partitioning nodes of the linked database into subsets according to a classification of the nodes;

forming a reduced link matrix whose elements represent links between the subsets of the partition;

calculating a block rank vector from the reduced link matrix;

selecting a component of the block rank vector corresponding to the block of nodes to obtain the rank value for the block of nodes.

28. The method of claim 27 wherein forming the reduced link matrix comprises computing a block link weight between a first block and a second block by adding together weights of links from nodes in the first block to nodes in the second block.